



# BÁO CÁO ĐỒ ÁN CUỐI KÌ

Sản phẩm: Hệ thống tưới nước thông minh



NHÓM THỰC HIỆN: NHÓM 5

GIÁO VIÊN HƯỚNG DẪN: CAO XUÂN NAM, ĐẶNG HOÀI THƯƠNG

## I. Danh sách thành viên nhóm:

### 1.Thông tin thành viên:

Nhóm 5

| Họ tên             | MSSV     |
|--------------------|----------|
| Huỳnh Thị Mỹ Thanh | 19127273 |
| Trần Thị Thùy Linh | 19127461 |
| Bùi Công Danh      | 19127348 |

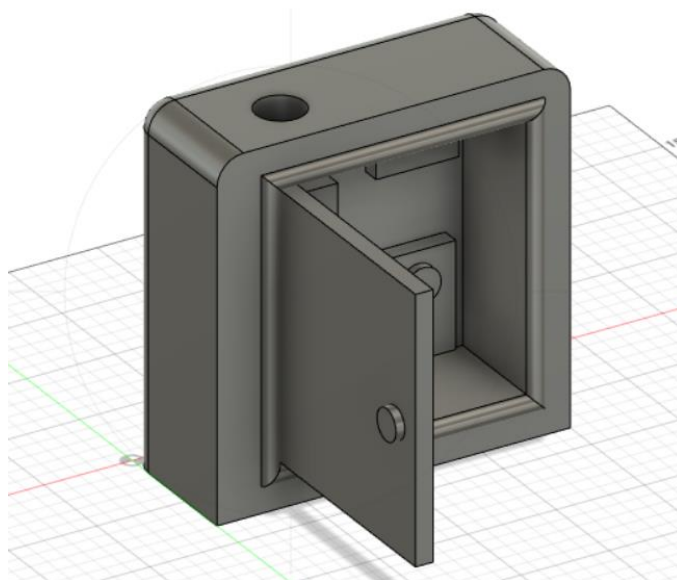
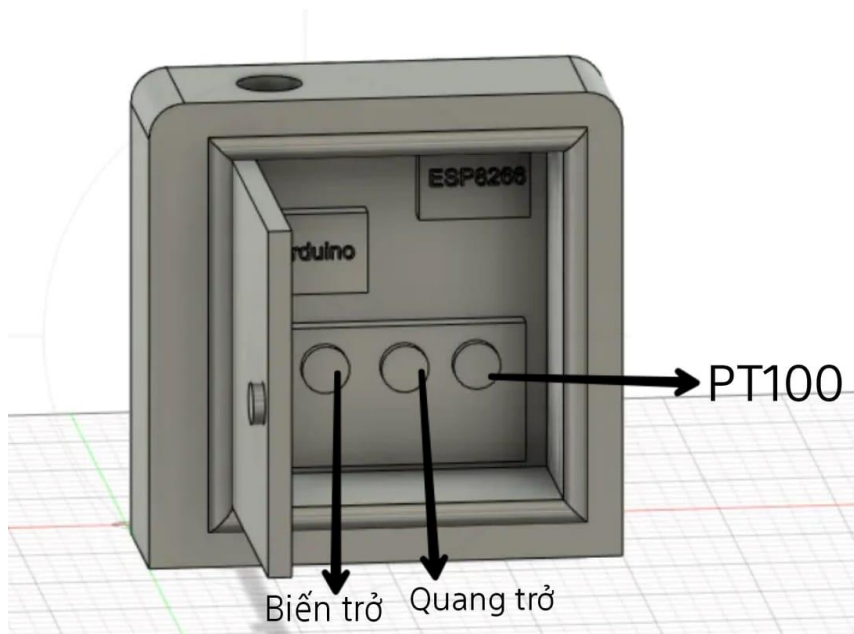
### 2.Bảng phân công công việc và đánh giá mức độ hoàn thành:

|                    | Công việc   | Mức độ hoàn thành |
|--------------------|---|-------------------|
| Huỳnh Thị Mỹ Thanh | Log in<br>Log out<br>Temp display (chart và gauge)<br>Gửi thông báo về điện thoại<br>Lưu dữ liệu trên thingspeak<br>Hiện thông báo đã tưới tự động trên website<br>Chỉnh sửa giao diện web<br>Report<br>Video demo                                | 100%              |
| Trần Thị Thùy Linh | SensorData<br>Brightness display (chart và gauge)<br>Điều khiển tưới bằng MQTT Explorer<br>Hiện thông báo đã tưới do MQTT Ex yêu cầu<br>Gửi thông báo về màn hình<br>Chỉnh sửa giao diện web<br>Report<br>Edit video demo                         | 100%              |
| Bùi Công Danh      | SensorData<br>Humidity display (chart và gauge)<br>Gửi thông báo lên màn hình<br>Gửi thông báo về điện thoại<br>Hiện thông báo đã tắt tưới tự động trên website<br>Lưu dữ liệu trên thingspeak<br>Chỉnh sửa giao diện web<br>Report<br>Edit video | 100%              |

## II. Tên sản phẩm:

Hệ thống tưới nước thông minh.

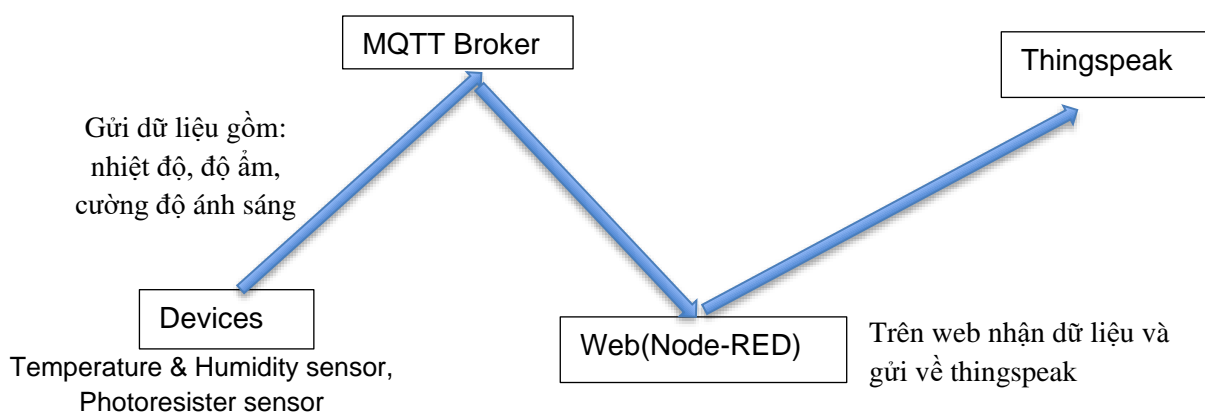
## III. Thiết kế 3D sản phẩm của nhóm:



#### IV. Mô tả chức năng của toàn hệ thống:

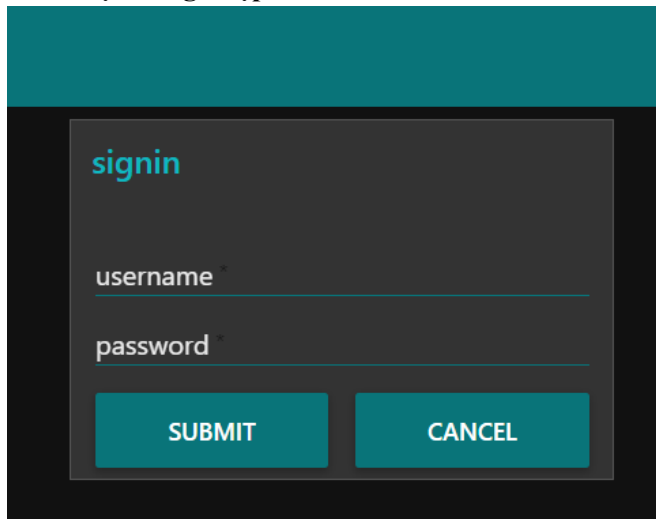
- ✚ Nhận các dữ liệu nhiệt độ, độ ẩm, cường độ ánh sáng vào (từ MQTT Explorer).
- ✚ Hiển thị các dữ liệu đã nhận được lên màn hình website.
- ✚ Từ các dữ liệu đã nhận được, cho phép người dùng quan sát dữ liệu hiện tại vừa được đẩy lên và quyết định có tưới nước hay không thông qua website có sẵn hai nút tự động tưới và tự động tắt.
- ✚ Người dùng có thể thông qua MQTT Explorer để yêu cầu hệ thống tưới nước cho cây.
- ✚ Lưu trữ các dữ liệu đã thu thập được trên thingspeak.
- ✚ Thông báo ra màn hình cũng như thông báo về điện thoại trong những trường hợp cấp bách (cần tưới nước và cần tắt nước gấp).
- ✚ Đăng nhập, đăng xuất tài khoản.

#### V. Sơ đồ truyền và nhận dữ liệu giữa các đối tượng trong hệ thống IoT:



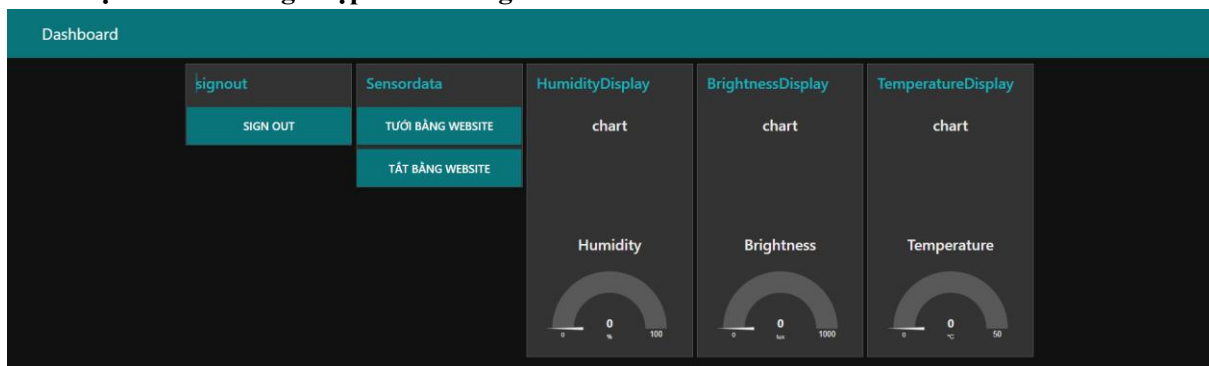
## VI. Giao diện Web và mô tả các chức năng của Web:

### Giao diện đăng nhập:

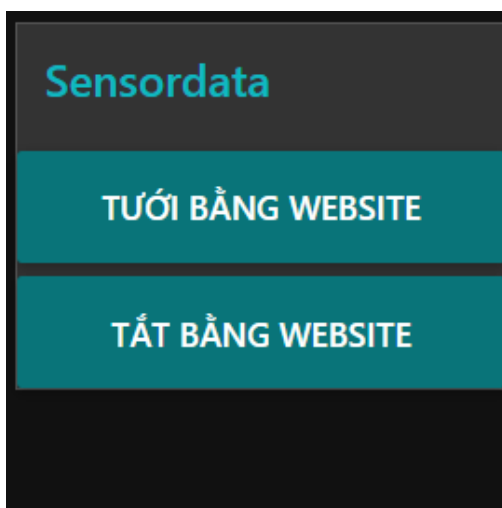
The login form is titled 'signin' in a light blue font. It features two input fields: 'username' and 'password', both with red asterisks indicating they are required. Below the fields are two buttons: 'SUBMIT' and 'CANCEL', both in light blue with white text. The form is set against a dark gray background.

Cho phép người dùng đăng nhập vào hệ thống vườn nhà mình.

### Giao diện sau khi đăng nhập thành công:



### SensorData là group gồm chức năng chính:



Điều khiển việc tưới nước thông qua Website. Khi click vào nút này, hệ thống sẽ tự động bật tưới nước cho cây.

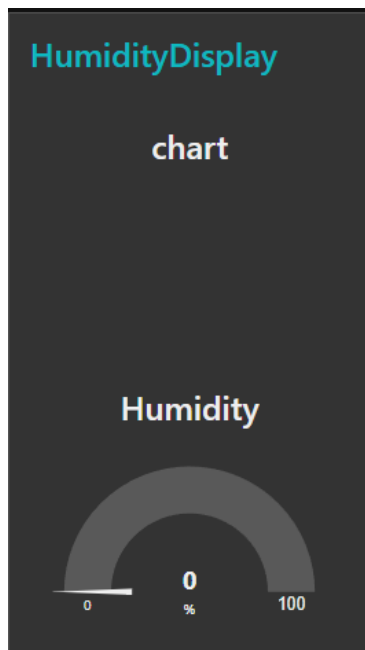


Tắt tưới nước thông qua Website. Khi người dùng muốn ngừng tắt nước bất kì lúc nào thì nút này sẽ đáp ứng yêu cầu này một cách nhanh chóng.



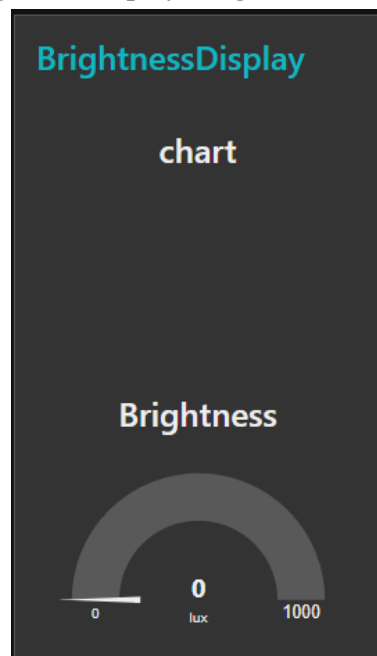
Điều khiển việc tưới cây thông qua điện thoại.

**HumidityDisplay là group có 2 chức năng chính:**



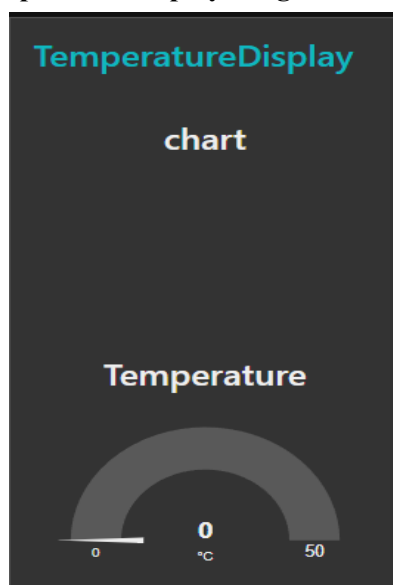
- ✚ Gauge: Hiển thị mức độ của độ ẩm (đơn vị phần trăm) để người dùng biết được tình hình độ ẩm hiện tại.
- ✚ Chart: Hiển thị biểu đồ đường về độ ẩm trong suốt khoảng thời gian theo dõi, giúp người dùng xem xét và tính toán các dự án khác trên cùng nông trại.

**BrightnessDisplay cũng có 2 chức năng chính:**



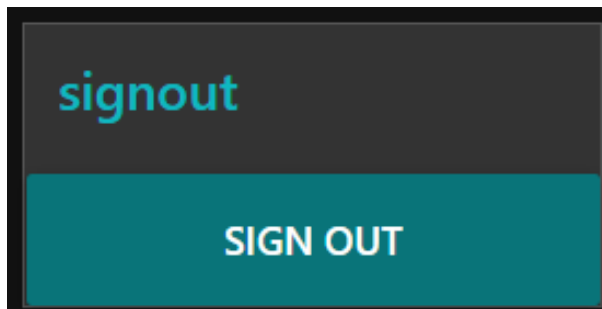
- ✚ Gauge: Hiển thị mức độ của độ sáng (lux) để người dùng biết được cường độ ánh sáng hiện tại.
- ✚ Chart: Hiển thị biểu đồ đường về cường độ ánh sáng trong thời gian theo dõi.

**TemperatureDisplay cũng có 2 chức năng chính:**



- ✚ Gauge: Hiển thị mức độ nhiệt độ hiện tại nhận được để người dùng nắm tình hình.
- ✚ Chart: Hiển thị biểu đồ đường về nhiệt độ trong thời gian theo dõi.

## Log out



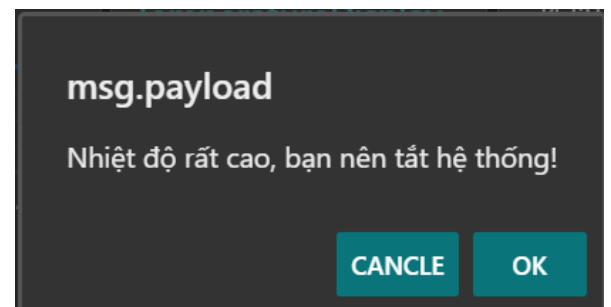
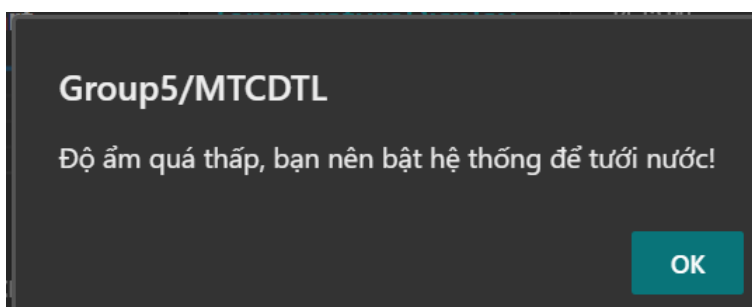
Hỗ trợ người dùng đăng xuất và trở về màn hình yêu cầu đăng nhập của website.

Ngoài những chức năng chính nêu trên, website còn có một số chức năng tự động như sau:

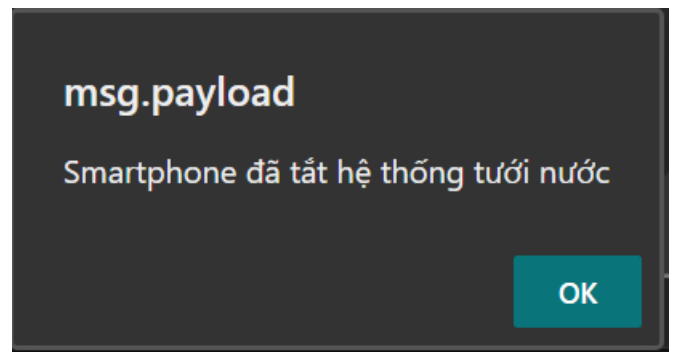
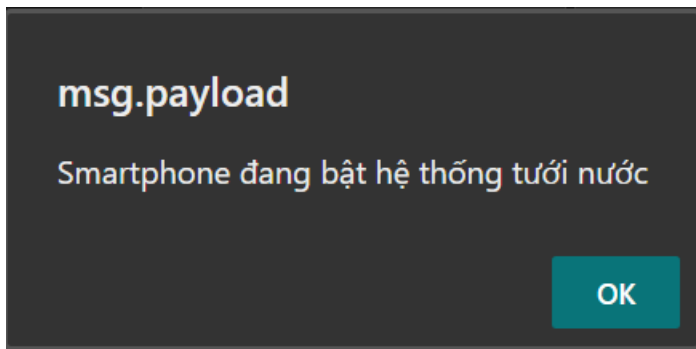
- 🌐 Gửi tin nhắn về điện thoại khi nhiệt độ quá cao, hoặc độ ẩm quá thấp:



- 🌐 Thông báo lên màn hình website nếu nhiệt độ quá cao, hoặc độ ẩm quá thấp:



🚦 Thông báo lên màn hình website nếu thiết bị đang điều khiển hệ thống tưới nước:



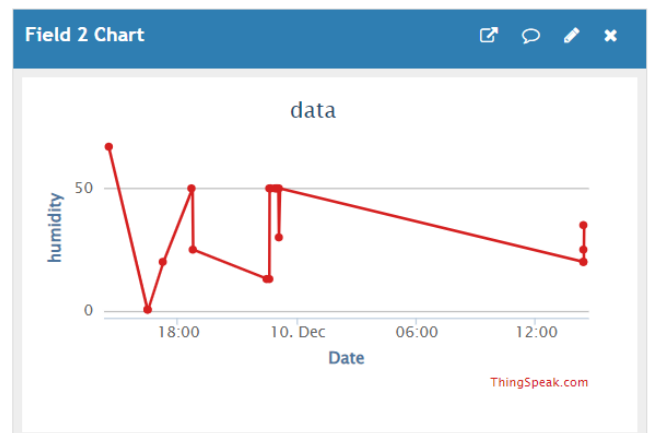
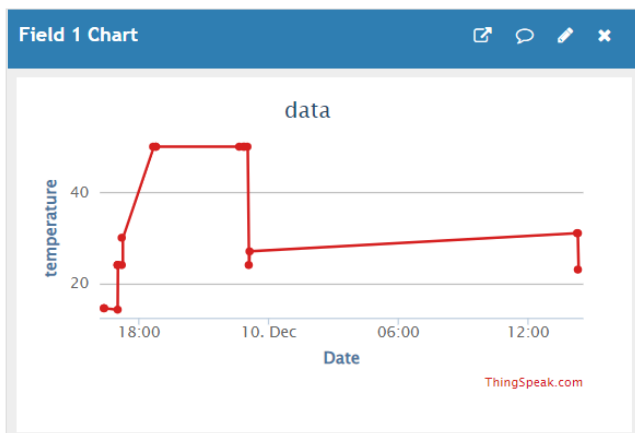
🚦 Lưu trữ dữ liệu website nhận được trên Thingspeak:

## Channel Stats

Created: [a.day.ago](#)

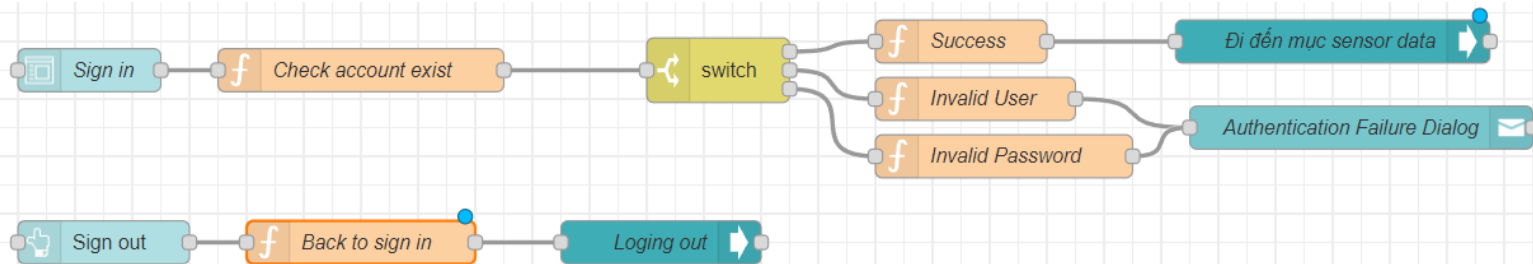
Last entry: [2.minutes.ago](#)

Entries: 39



## VII. Node-red:

### 1. Sign in, Sign out



## A. SIGN IN

### Property của node sign in (form):

Group: [Dashboard] signin

Size: auto

Label: optional label

| Label    | Name     | Type     | Required                            | UiRows | Remove |
|----------|----------|----------|-------------------------------------|--------|--------|
| username | username | Text     | <input checked="" type="checkbox"/> |        |        |
| password | password | Password | <input checked="" type="checkbox"/> |        |        |

Gồm có 2 đầu vào chính:

1. Username: (htmthanh, bcdanh, ttlinh...) ở định dạng text.
2. Password: (htmthanh, bcdanh, ttlinh...) ở định dạng password.

### Property của node Check account exist (kiểm tra tài khoản có tồn tại hay không):

```
1 var accounts = flow.get("accounts") || [ { username : "htmthan  
2  
3 var username = msg.payload.username ;  
4 var password = msg.payload.password ;  
5  
6 var nMsg = {}; nMsg.payload = 2;  
7  
8 accounts.forEach(function ( account ){  
9   if ( account.username == username ) {  
10     nMsg.payload = 3;  
11     if ( account.password == password ) {  
12       nMsg.payload = 1;  
13     }  
14   }  
15 });  
16  
17 return nMsg;
```

Lưu một loạt tài khoản vào accounts.  
Dùng 2 biến username, password (khi người dùng nhập vào) để kiểm tra.  
Mặc định giá trị của node này trả về ban đầu là 2 (sai username).  
Kiểm tra nếu username đúng, password sai => trả về 3.  
Kiểm tra nếu username đúng, password đúng => trả về 1.



### Property của node switch:



- 🔗 Nếu = 1 : Đăng nhập thành công.  
Cài đặt node success với property như sau:

```
1
2 msg.payload = {
3   "group": {
4     "show": ["Dashboard_Sensordata", "Dashboard_signout",
5     "Dashboard_TemperatureDisplay", "Dashboard_BrightnessDisplay", "
6     "hide": ["Dashboard_signin"],
7   },
8 },
9   "tab": "Dashboard"
10 }
11
12 return msg;
```

Cho hiển thị các chức năng của website lên màn hình, đồng thời tắt group sign in đi.

- 🔗 Nếu = 2 : Sai username.  
Cài đặt node invalid user với property:

```
msg.payload = "Invalid username!"
return msg;
```

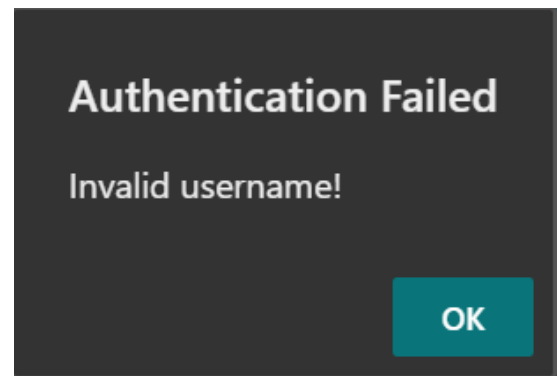
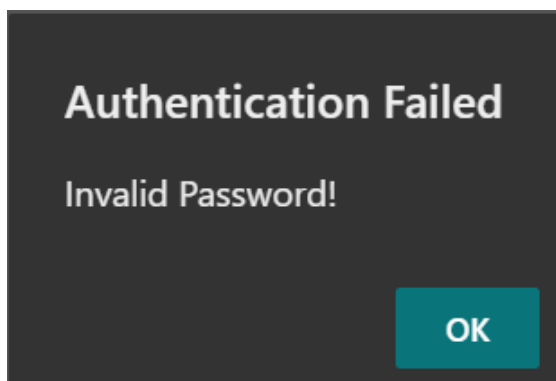
Để gửi thông báo “Invalid username” ra màn hình thông qua node notification.

- 🔗 Nếu = 3 : Sai password.  
Cài đặt node invalid password với property:

```
1 msg.payload = "Invalid Password!";
2 return msg;
```

Để gửi thông báo “Invalid password” ra màn hình thông qua node notification.

**Node notification: in thông báo ra màn hình.**



## B.SIGN OUT

Gồm button (Sign out), function (Back to sign in), ui control (logging out)

Property node function:

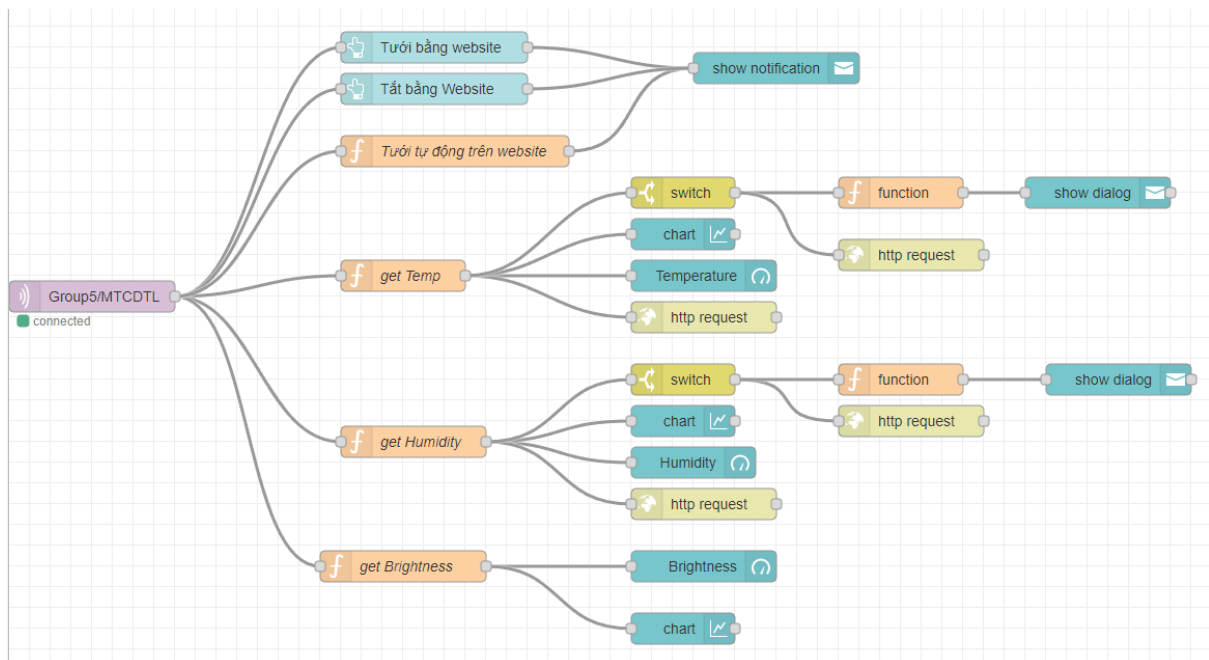
```
1 var msg = {};  
2 msg.payload = {  
3   "group": {  
4     "hide": ["Dashboard_Sensordata", "Dashboard_signout",  
5             "Dashboard_TemperatureDisplay", "Dashboard_BrightnessDi  
6     "show": ["Dashboard_signin"],  
7   },  
8   "tab": "Dashboard"  
9 };  
10 return msg;
```

Tắt màn hình hiện tại  
và trở về trạng thái yêu  
cầu đăng nhập ban đầu

## 2.SensorData và các group display:

Node MQTT in để hỗ trợ giả lập đưa dữ liệu vào website.

### A.SENSORDATA



Property của button tưới bằng website:

☒ When clicked, send:

Payload

Topic

Set giá trị payload để truyền thông báo ra màn hình.

**Hệ thống đang tưới nước**

### Property của button tắt bằng website:

✉ When clicked, send:

Payload

Topic

Tương tự như node tưới bằng website, truyền giá trị vào payload để trả ra màn hình thông báo.



### Cấu hình node tưới tự động do website thiết lập:

```
1 var p = msg.payload;
2 const myArr = p.split(" ");
3
4 if(parseFloat(myArr[0]) <= 25&& parseFloat(myArr[1]) < 30&&parseFloat(myArr[2]) < 50)
5 {
6   msg.payload = "Hệ thống đang tưới nước";
7 }
8 else
9 {
10  msg.payload = "Hệ thống đang tắt";
11 }
12
13 return msg;
```

Có 3 ngưỡng giá trị đặt ra ban đầu:

Nhiệt độ <25, độ ẩm <30, độ sáng <50 => cho hệ thống tự động tưới cây.

Ngược lại, tự động tắt hệ thống.

### B.Temperature Display:

#### Cấu hình node get Temp:

```
1 var p = msg.payload;
2 const myArr = p.split(" ");
3 msg.payload = parseFloat(myArr[0]);
4 return msg;
```

Khi nhận 1 dữ liệu sẽ gồm 3 số: <nhiệt độ> <độ ẩm> <ánh sáng>

Nên sẽ chia thành 3 phần tử riêng biệt bởi hàm split(""). Ép kiểu phần tử đầu (nhiệt độ) về float và trả về.

### Node switch:

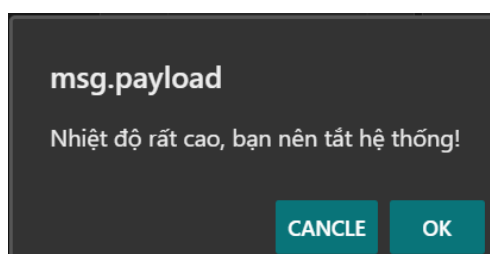
☰

Nếu nhiệt độ từ 30 độ trở đi sẽ thông báo qua màn hình website và gửi về điện thoại cảnh báo.

+Node function: Set giá trị cho payload rồi thông báo về màn hình web qua node notification

```
1 msg.payload="Nhiệt độ rất cao, bạn nên tắt hệ thống! "
2 return msg;
```

Set giá trị cho payload rồi thông báo về màn hình web qua node notification:



+Node http request: gửi thông báo về điện thoại thông qua link IFTTT (dán vào URL trong property).

**Node chart, gauge (temperature):** Hiển thị biểu đồ đường (chart), mức độ của nhiệt độ (gauge).

**Node http request (cuối cùng):** gửi dữ liệu lên Thingspeak thông qua đường link.

### C.Humidity Display:

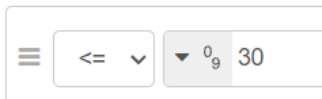
**Cấu hình node get humidity:**

```
1 var p = msg.payload;
2 const myArr = p.split(" ");
3 msg.payload = parseFloat(myArr[1]);
4 return msg;
```

Tương tự như node get temp, ép kiểu phần tử thứ 2 (độ ẩm) về float và trả ra.

### Node switch:

Nếu độ ẩm <30% thì hệ thống sẽ thông báo lên màn hình website và gửi về điện thoại để thông báo.



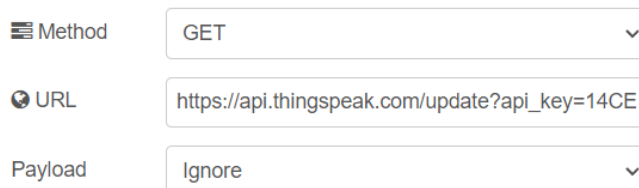
+Node function: set giá trị cho payload để hiển thị về màn hình.

```
msg.payload="Độ ẩm quá thấp, bạn nên bật hệ thống để tưới nước! "
return msg;
```

+Http request: gửi tin nhắn thông báo về điện thoại thông qua link IFTTT.

**Node chart, gauge (humidity):** Hiển thị biểu đồ đường (chart), mức độ của độ ẩm (gauge).

**Node http request (cuối cùng):** gửi dữ liệu lên Thingspeak thông qua đường link.



### D.Brightness Display:

**Cấu hình node get brightness:**

Tương tự như node get temp, ép kiểu phần tử thứ 3 (cường độ ánh sáng) về float và trả ra.

```
1 var p = msg.payload;
2 const myArr = p.split(" ");
3 msg.payload = parseFloat(myArr[2]);
4 return msg;
```

**Node chart, gauge (brightness):** Hiển thị biểu đồ đường (chart), mức độ của độ sáng (gauge).

Ở phần brightness display này, không cần lưu trữ dữ liệu lên thingspeak cũng như thông báo về điện thoại.

### 3. Nhận tín hiệu từ thiết bị và in ra thông báo:



#### Cấu hình node mqtt in để nhận tín hiệu từ thiết bị:

**Edit mqtt in node**

Delete Cancel Done

**Properties**

Server: broker.mqtt-dashboard.com:1883

Action: Subscribe to single topic

Topic: Group5/Devices

Server và port của mqtt

Topic subscribe

#### Cấu hình node function:

```
1 if (msg.payload==1)
2   msg.payload="Smartphone đang bật hệ thống tưới nước ";
3 else
4   msg.payload="Smartphone đã tắt hệ thống tưới nước ";
5 return msg;
```

Kiểm tra tín hiệu từ thiết bị (được giả lập gửi bằng mqtt explorer) được nhận bởi node mqtt in . Nếu tín hiệu là 1 thì thiết bị đang bật hệ thống tưới nước , còn lại là thiết bị đang tắt hệ thống tưới nước.

#### Cấu hình node notification: để in câu thông báo nhận được ở node function dưới dạng dialog

**Edit notification node**

Delete Cancel Done

**Properties**

Layout: OK / Cancel Dialog

Send to all browser sessions. ☒